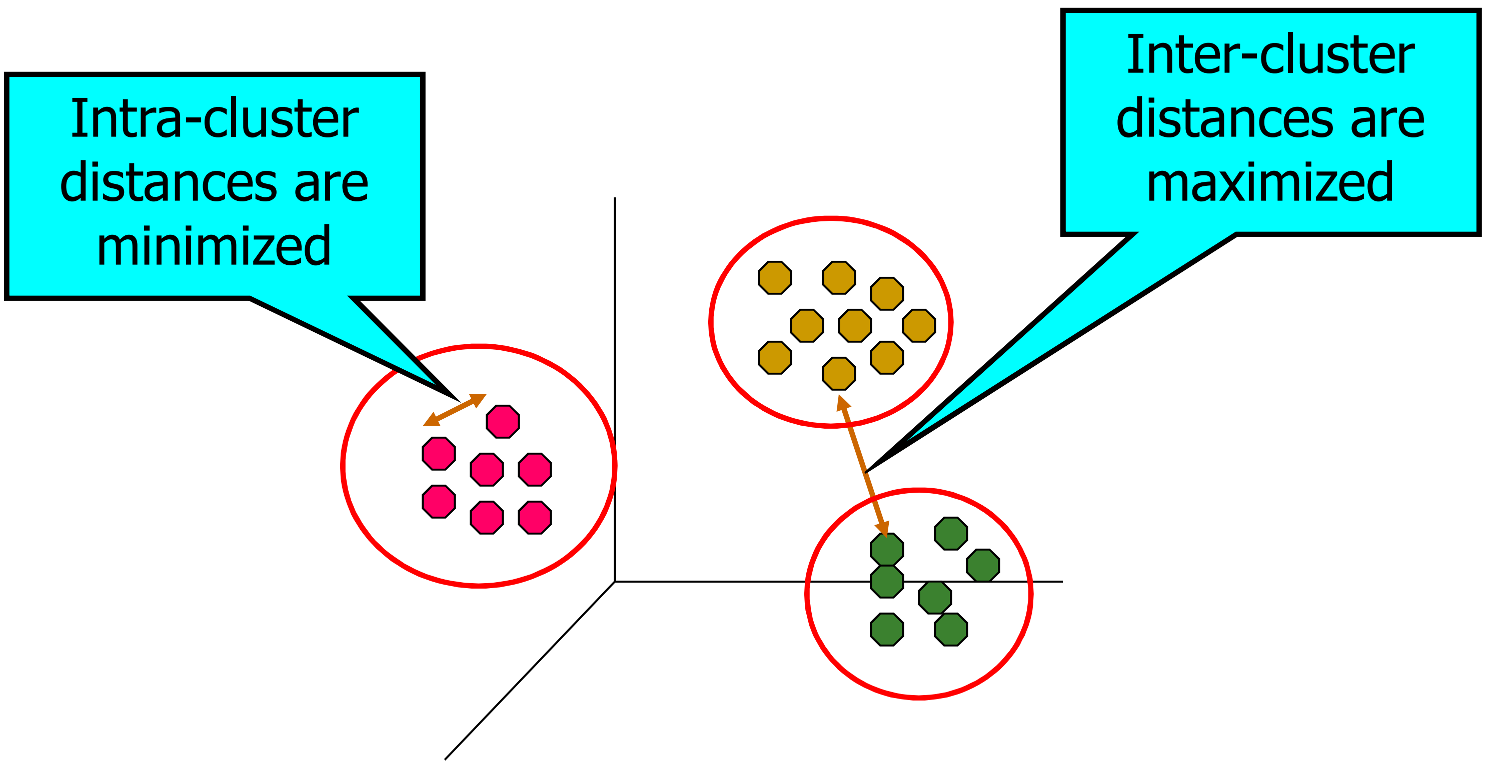
Computer Science 477

Basic Clustering

Lecture 14

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



General Theme

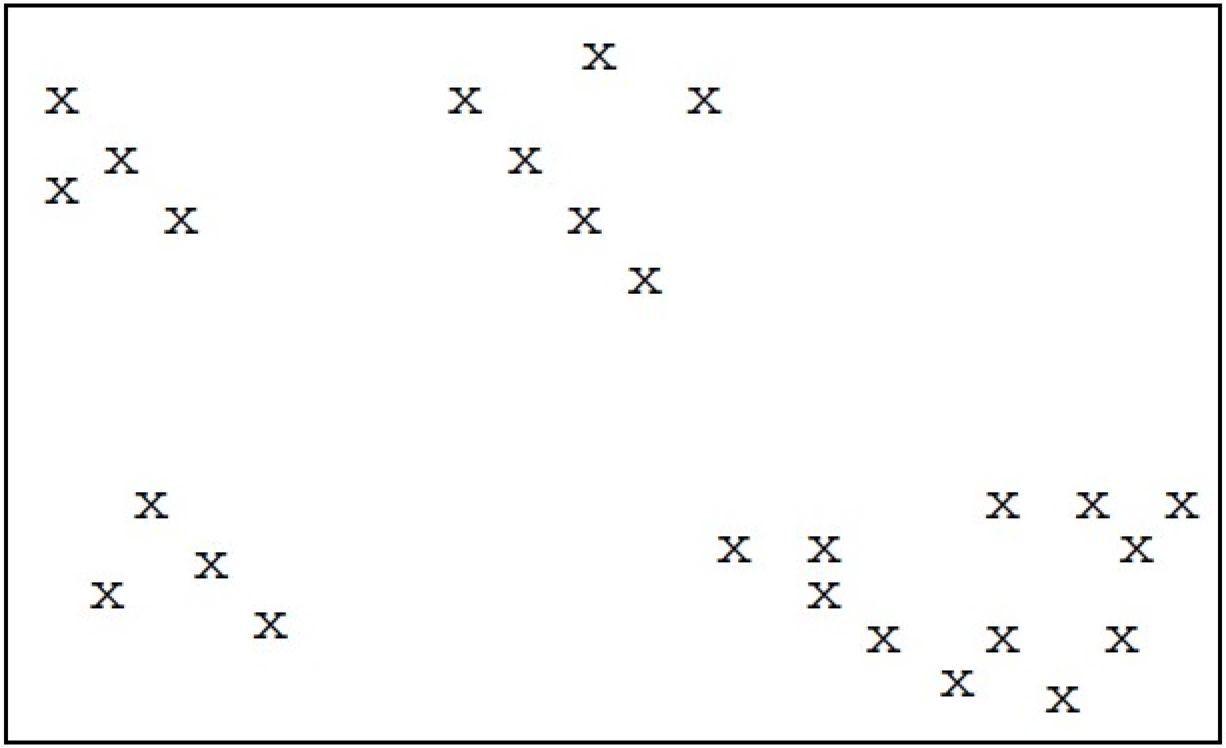
- Extracting information from unlabeled data and turn to the important topic of *clustering*.
- Clustering concerned with grouping together objects that are
  - Similar to each other
  - Dissimilar to the objects belonging to other clusters.
- Here: two methods for which the similarity between objects is based on a measure of the distance between them
  - Two among many methods
- In data exploration, cluster can be preliminary to classification

Clustering Applications

- In an economics application we might be interested in finding countries whose economies are similar.
- In a financial application we might wish to find clusters of companies that have similar financial performance.
- In a marketing application we might wish to find clusters of customers with similar buying behavior.
- In a medical application we might wish to find clusters of patients with similar symptoms.
- In a document retrieval application we might wish to find clusters of documents with related content.
- In a crime analysis application we might look for clusters of high volume crimes such as burglaries or try to cluster together much rarer (but possibly related) crimes such as murders.

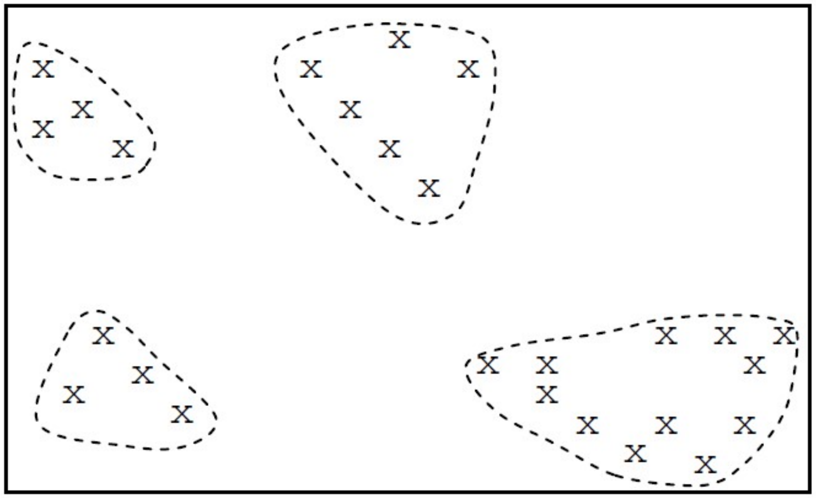
Restricted Case

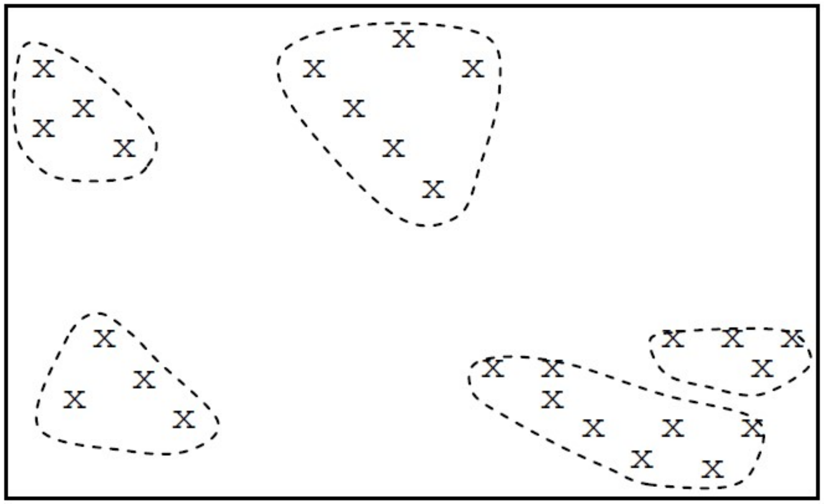
- Where there are only two attributes, can be visualized as a plot on an x,y plane



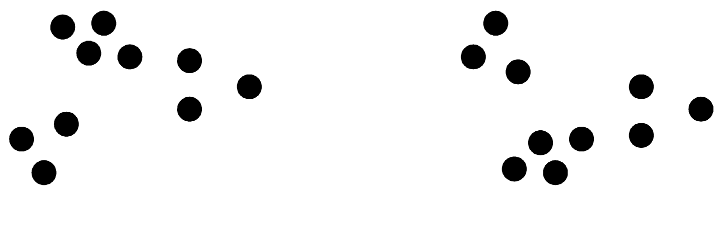
Visualization

- One cluster or two?



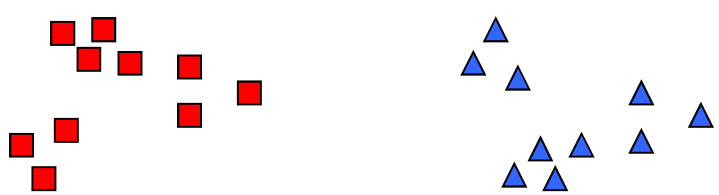


Cluster Analysis is not unequivocal



How many clusters?

Six Clusters



Two Clusters

Four Clusters

Groundwork

- Assume that all attribute values are continuous
  - If they are not, recall Chapter 2
- Need the notion of the 'center' of a cluster, generally called its *centroid.*
- Assuming that we are using Euclidean distance or something similar as a measure
  - Centroid of a cluster to be the point for which each attribute value is the average of the values of the corresponding attribute for all the points in the cluster.
- So the centroid of the four points (with 6 attributes)

8.0	7.2	0.3	23.1	11.1	−6.1
2.0	−3.4	0.8	24.2	18.3	−5.2
−3.5	8.1	0.9	20.6	10.2	−7.3
−6.0	6.7	0.5	12.5	9.2	−8.4

will be

0.125	4.65	0.625	20.1	12.2	−6.75

Types of Clustering

- A clustering is a set of clusters

- Important distinction between hierarchical and partitional sets of clusters

- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset

- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Types of Clusters

- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual
- Described by an Objective Function

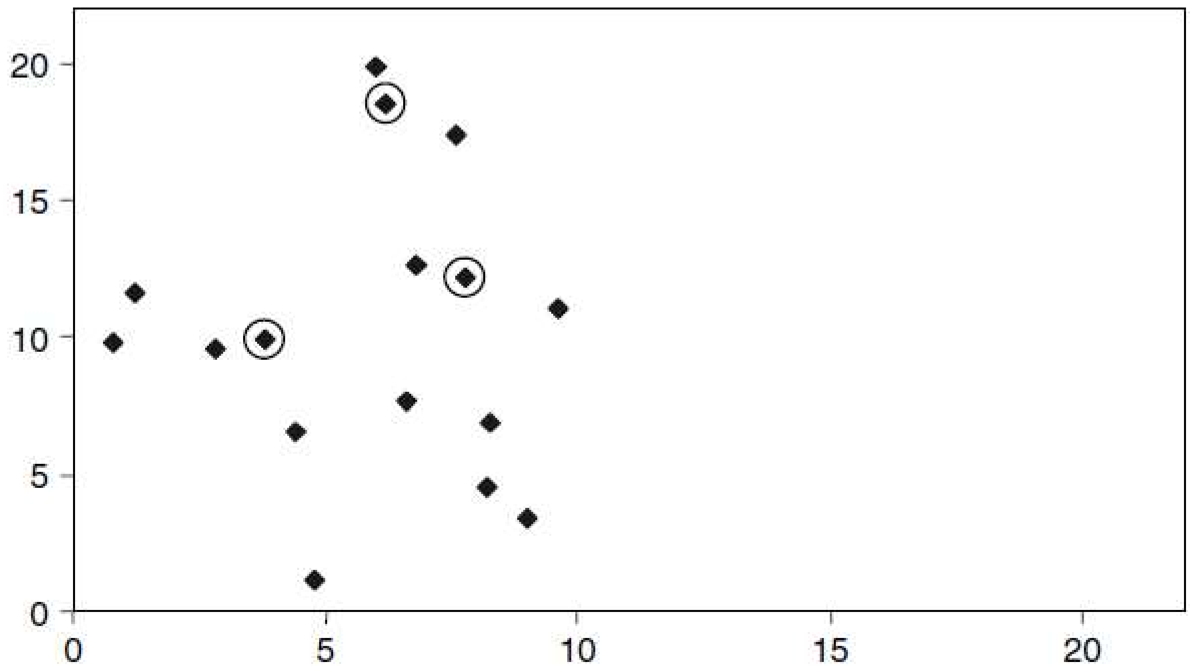
k-Means Clustering

- Set a value for the number of clusters
  - *K,* generally a small integer (2,3,4,5)
- Choose *k* points as initial centroids
  - (generally corresponding to the location of *k* of the objects).
- Chose points far apart, generally.
- Assign each instance to a cluster
  - Calculating the nearest centroid.
- Recalculate the centroids of the clusters
- Repeat the assignment of each instance to the most recently calculated centroid.

K-means Clustering – Details

- Initial centroids are often chosen randomly.
  - Clusters produced vary from one execution run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, correlation, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is O(n * K * I * d)
  - n = number of points, K = number of clusters,
I = number of iterations, d = number of attributes

K-means Example



Objects For Clustering

x	y
6.8	12.6
0.8	9.8
1.2	11.6
2.8	9.6
3.8	9.9
4.4	6.5
4.8	1.1
6.0	19.9
6.2	18.5
7.6	17.4
7.8	12.2
6.6	7.7
8.2	4.5
8.4	6.9
9.0	3.4
9.6	11.1

Select $k = 3$

Circled points initial centroids

	Initial	
	x	y
Centroid 1	3.8	9.9
Centroid 2	7.8	12.2
Centroid 3	6.2	18.5

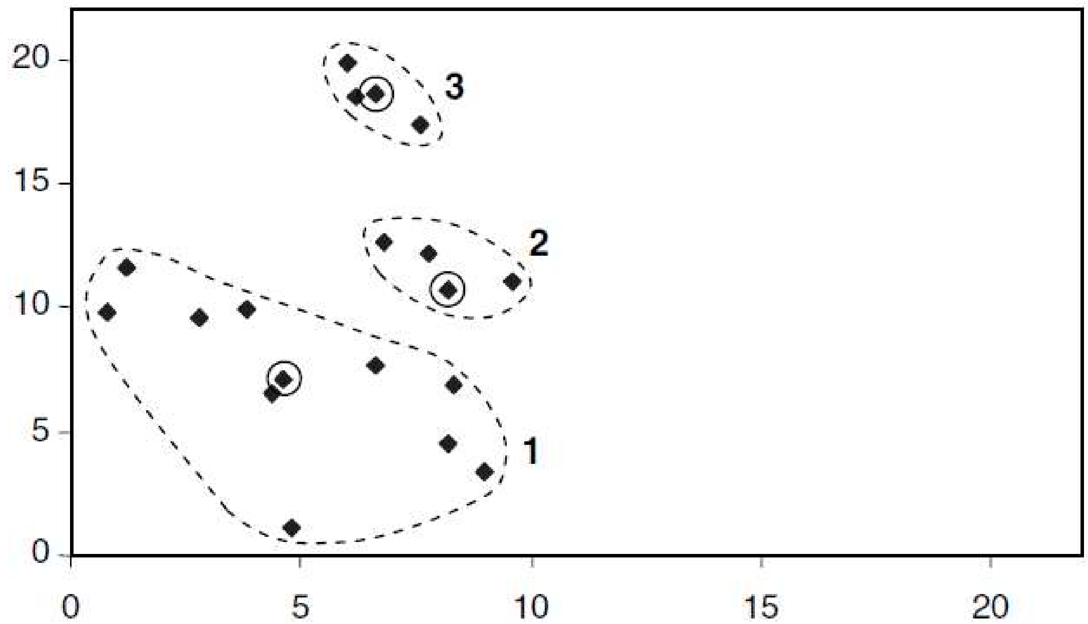
Example – Recalculating Centroids

- Colums labeled d1, d2, d3 give the distance of each point from initial centroids (d1, d2, d3)
- Simple Euclidian distance
- First distance calculated

$\sqrt{(6.8 - 3.8)^2 + (12.6 - 9.9)^2} = 4.0$

x	y	$d1$	$d2$	$d3$	cluster
6.8	12.6	4.0	1.1	5.9	2
0.8	9.8	3.0	7.4	10.2	1
1.2	11.6	3.1	6.6	8.5	1
2.8	9.6	1.0	5.6	9.5	1
3.8	9.9	0.0	4.6	8.9	1
4.4	6.5	3.5	6.6	12.1	1
4.8	1.1	8.9	11.5	17.5	1
6.0	19.9	10.2	7.9	1.4	3
6.2	18.5	8.9	6.5	0.0	3
7.6	17.4	8.4	5.2	1.8	3
7.8	12.2	4.6	0.0	6.5	2
6.6	7.7	3.6	4.7	10.8	1
8.2	4.5	7.0	7.7	14.1	1
8.4	6.9	5.5	5.3	11.8	2
9.0	3.4	8.3	8.9	15.4	1
9.6	11.1	5.9	2.1	8.1	2

Initial Clusters

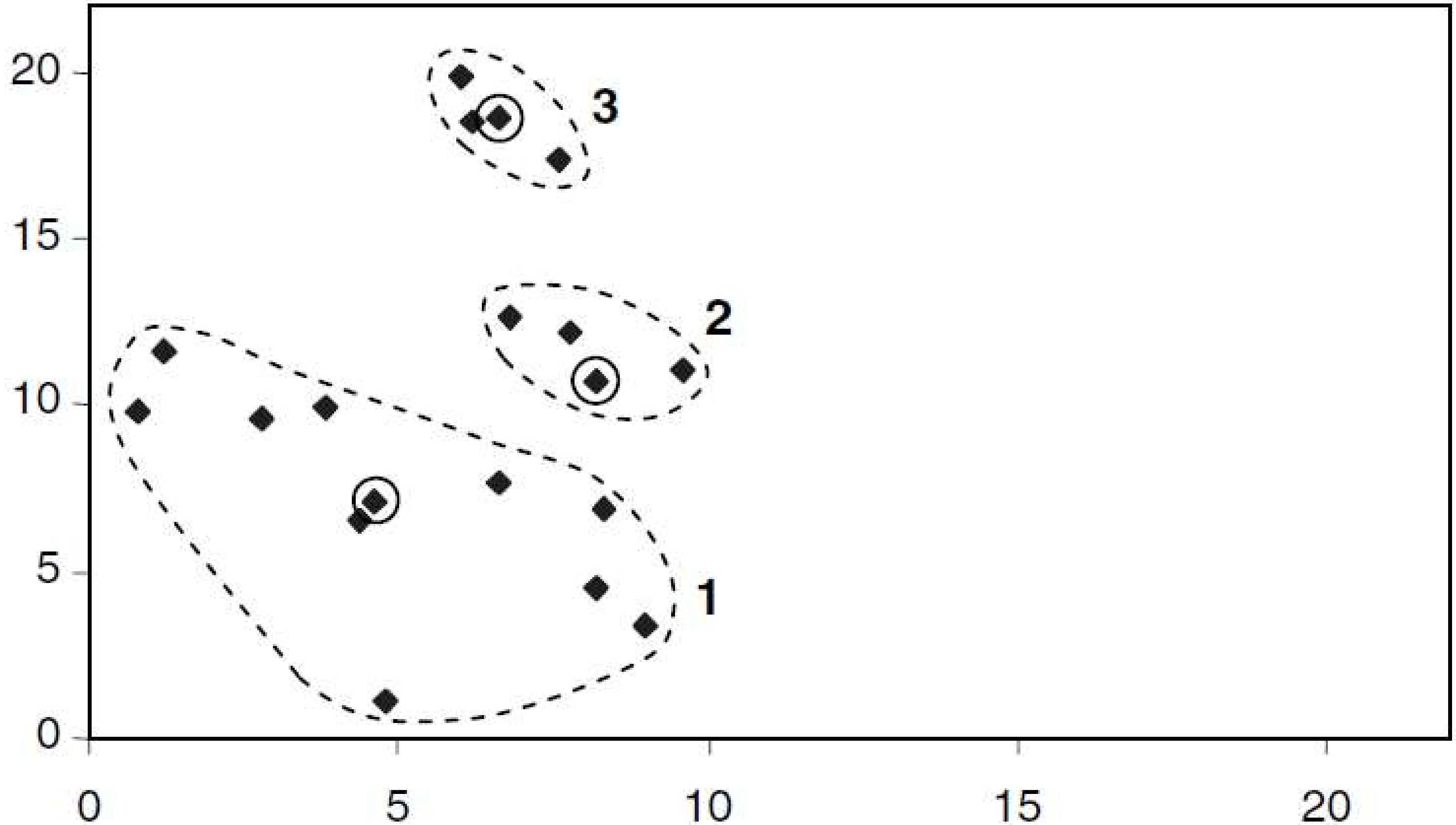


- Recalculate centroids

	Initial		After first iteration	
	x	y	x	y
Centroid 1	3.8	9.9	4.6	7.1
Centroid 2	7.8	12.2	8.2	10.7
Centroid 3	6.2	18.5	6.6	18.6

Recluster

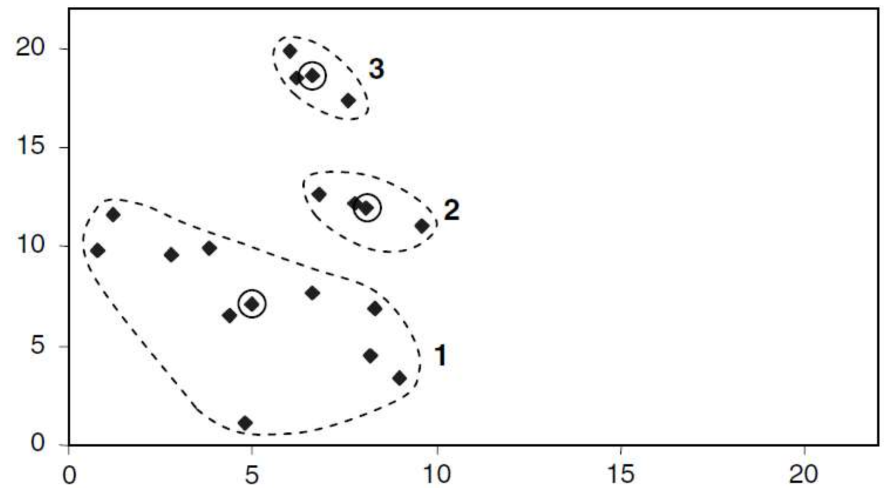
- Reassign all points to (possibly) new clusters
  - Closer to new centroids



Recalculate Centroids and Recluster Again

- The first two centroids have moved a little, but the third has not moved at all.

	Initial		After first iteration		After second iteration	
	x	y	x	y	x	y
Centroid 1	3.8	9.9	4.6	7.1	5.0	7.1
Centroid 2	7.8	12.2	8.2	10.7	8.1	12.0
Centroid 3	6.2	18.5	6.6	18.6	6.6	18.6



- Third clustering
- Centroids have not moved, so we are done

Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
- For each point, the error is the distance to the nearest cluster
- To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

- x is a data point in cluster C_i and m_i is the representative point for cluster C_i
  - Can show that m_i corresponds to the center (mean) of the cluster
- Given two clusters, we can choose the one with the smallest error
- One easy way to reduce SSE is to increase K, the number of clusters
  - A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

Results

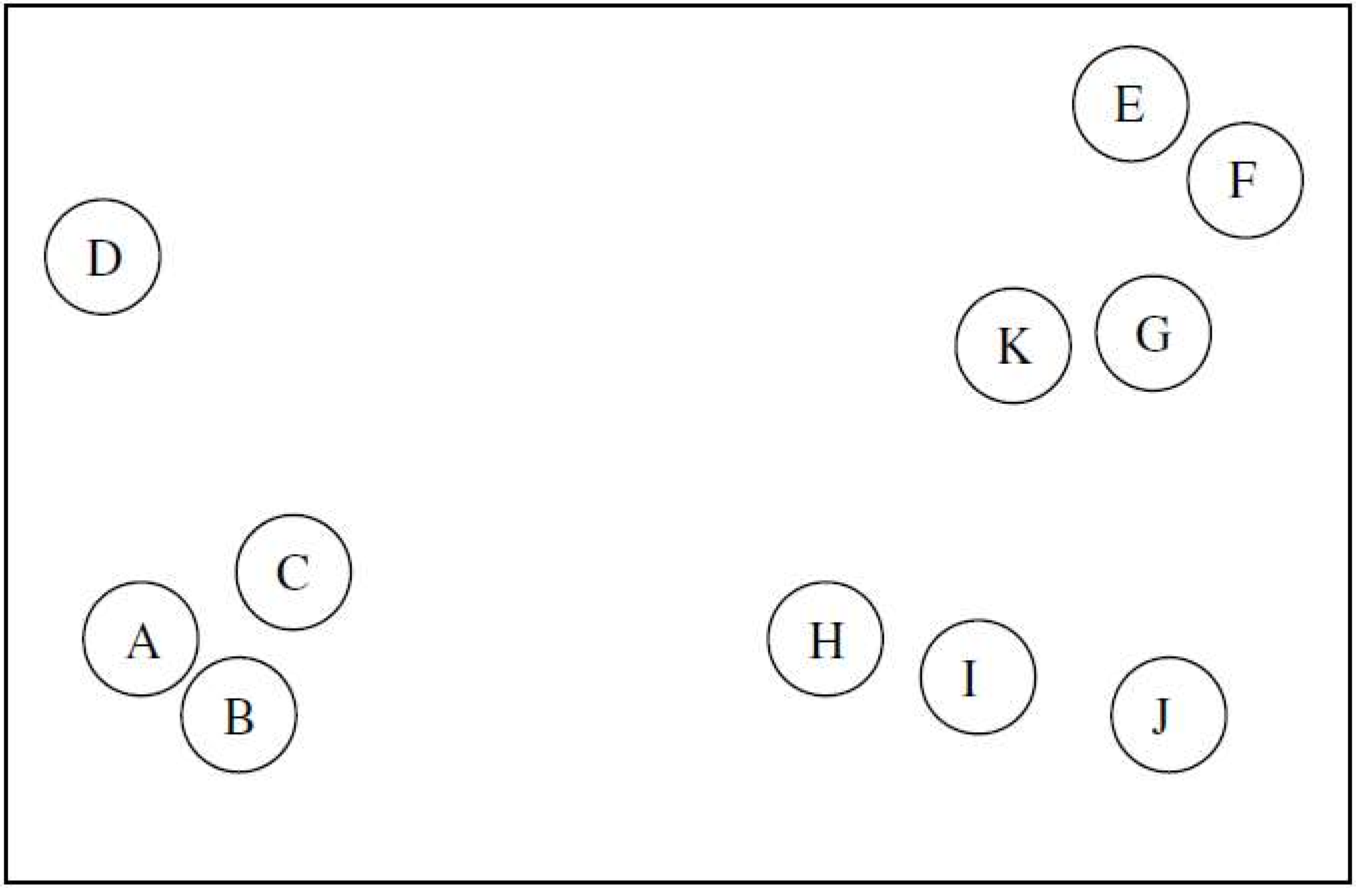
Value of k	Value of objective function
1	62.8
2	12.3
3	9.4
4	9.3
5	9.2
6	9.1
7	9.05

- Results suggest that the best value of *k* is probably 3.
  - Value of the function for *k* = 3 is much less than for *k* = 2, but only a little better than for *k* = 4.
- It is possible that the value of the objective function drops sharply after *k* = 7,
  - *K*=3 still preferred.
  - Small number of clusters as far as possible.
- *Not* trying to find the value of *k* with the smallest value of the objective function.
- That will occur when the value of *k* is the same as the number of objects
  - Each object forms its own cluster of one.
  - Objective function will then be zero, but the clusters will be worthless.

Agglomerative Hierarchical Clustering

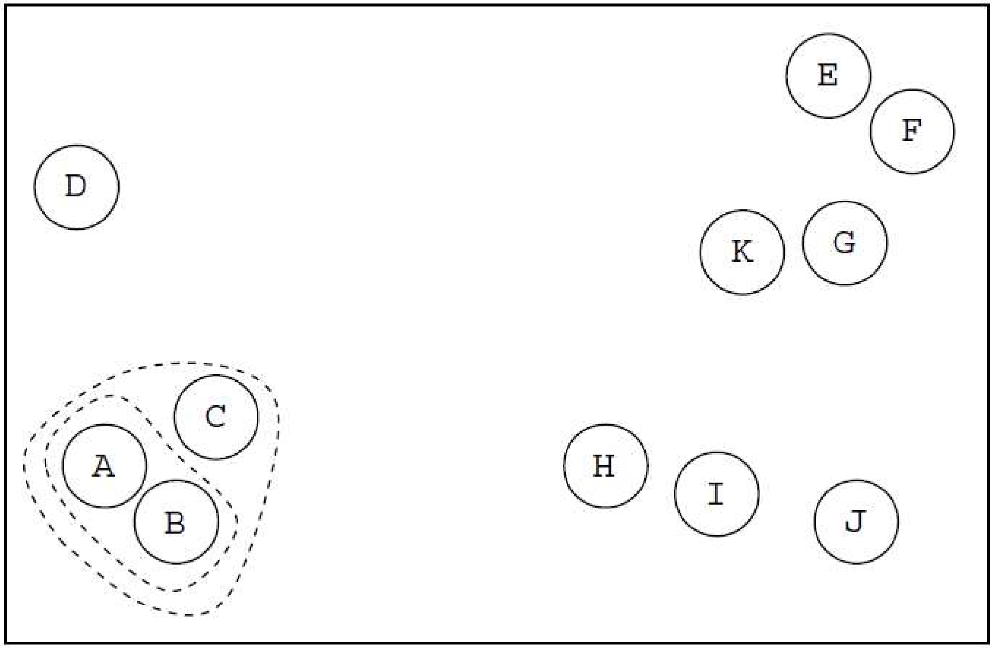
- Start with each object in a cluster of its own and then repeatedly merge the closest pair of clusters until we end up with just one cluster containing everything.
- Algorithm:
  - 1. Assign each object to its own single-object cluster.
    - Calculate the distance between each pair of clusters.
  - 2. Choose the closest pair of clusters and merge them into a single cluster
    - (so reducing the total number of clusters by one).
  - 3. Calculate the distance between the new cluster and each of the old clusters.
  - 4. Repeat steps 2 and 3 until all the objects are in a single cluster.

Initial State



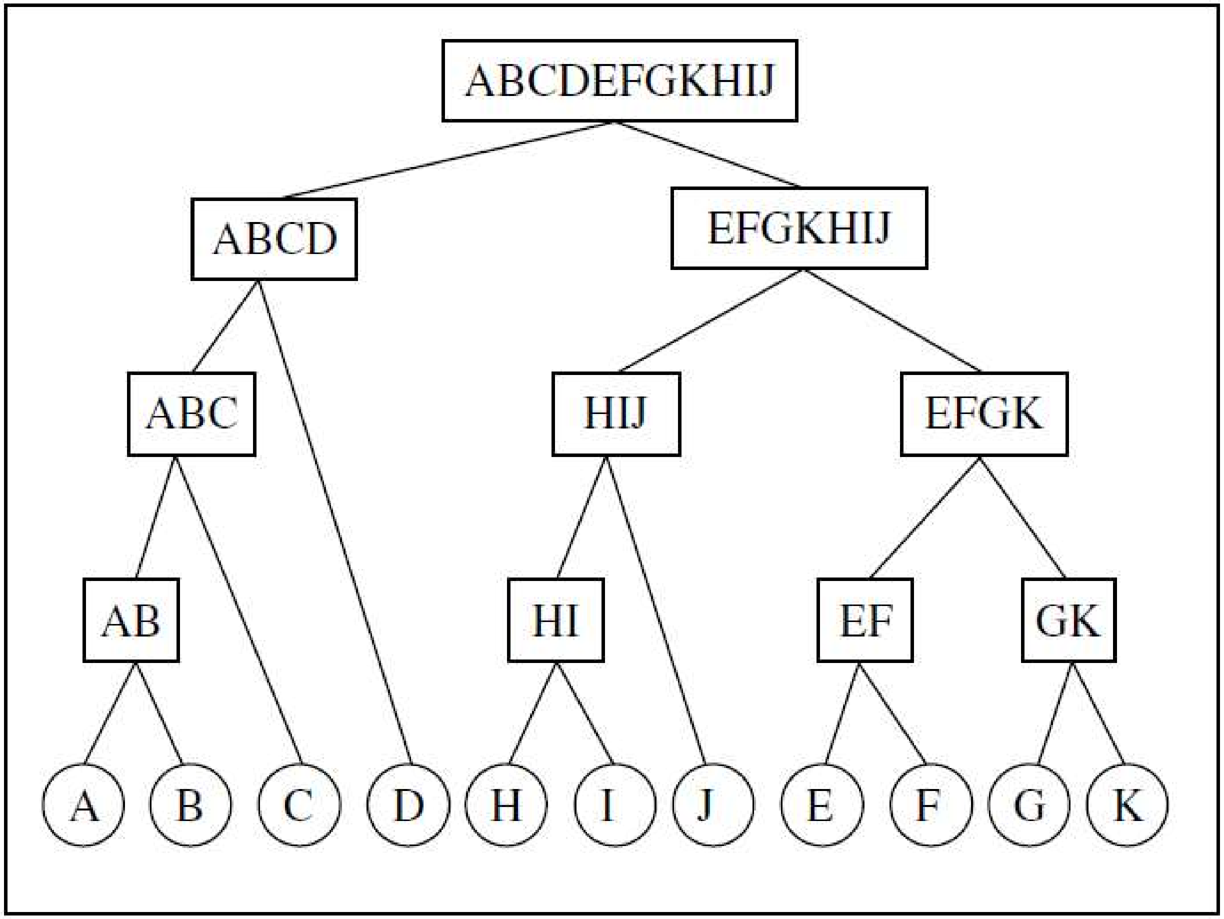
- Initially, every data point constitutes it own cluster

Sequence of Merges



- 1. A and B → AB
- 2. AB and C → ABC
- 3. G and K → GK
- 4. E and F → EF
- 5. H and I → HI
- 6. EF and GK → EFGK
- 7. HI and J → HIJ
- 8. ABC and D → ABCD
- 9. EFGK and HIJ → EFGKHIJ
- 10. ABCD and EFGKHIJ → ABCDEFGKHIJ

Dendrogram – Agglomeration History



- Tree of successive mergers

Distance between Clusters

- No need to recalculate cluster distances at each iteration
  - Only distances that change are among most recently merged
- Maintain a distance matrix
- Initially, an entry for each data point

	a	b	c	d	e	f
a	0	12	6	3	25	4
b	12	0	19	8	14	15
c	6	19	0	12	5	18
d	3	8	12	0	11	9
e	25	14	5	11	0	7
f	4	15	18	9	7	0

  - Symmetric
  - Diagonals zero

Distance Measures

- Might use cluster centroids to define cluster distance
- *Single-link clustering* the distance between two clusters is shortest distance from any member of one cluster to any member of the other cluster.
  - On this measure the distance from *ad* to *b* is 8
  - The shorter of the distance from *a* to *b* (12) and the distance from *d* to *b* (8) in the original distance matrix

	a	b	c	d	e	f
a	0	12	6	3	25	4
b	12	0	19	8	14	15
c	6	19	0	12	5	18
d	3	8	12	0	11	9
e	25	14	5	11	0	7
f	4	15	18	9	7	0

- Two alternatives to *single-link clustering* are *complete-link clustering* and *average-link clustering*
- Distance between two clusters the longest distance from any member of one cluster to any member of the other cluster, or the average such distance respectively.